intel

APPLICATION NOTE

AP-271

April 1986

Applying The 5C121 Architecture

JIM DONNELL
PROGRAMMABLE LOGIC APPLICATIONS
INTEL CORPORATION

Order Number: 292008-001

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

The following are trademarks of Intel Corporation and may only be used to identify Intel Products:

Above, BITBUS, COMMputer, CREDIT, Data Pipeline, GENIUS, i, î, ICE, iCEL, iCS, iDBP, iDIS, I²ICE, iLBX, i_m, iMDDX, iMMX, Insite, Intel, int_el, int_elBOS, Intelevision, $int_eligent$ Identifier, $int_eligent$ Programming, Intellec, Intellink, iOSP, iPDS, iPSC, iRMX, iSBC, iSBX, iSDM, iSXM, KEPROM, Library Manager, MAP-NET, MCS, Megachassis, MICROMAINFRAME, MULTIBUS, MULTICHANNEL, MULTIMODULE, ONCE, OpenNET, PC-BUBBLE, Plug-A-Bubble, PROMPT, Promware, QUEST, QueX, Quick-Pulse Programming Ripplemode, RMX/80, RUPI, Seamless, SLD, UPI, and VLSiCEL, and the combination of ICE, iCS, iRMX, iSBC, iSBX, iSXM, MCS, or UPI and a numerical suffix 4-SITE.

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

*MULTIBUS is a patented Intel bus.

Additional copies of this manual or other Intel literature may be obtained from:

Intel Literature Inquiries
SC6-58
P.O. Box 58065
Santa Clara, CA 95052-8065

APPLYING THE 5C121 ARCHITECTURE

CONTENTS

INTRODUCTION

Intel's 5C121 Erasable Programmable Logic Device represents a new breed in the world of programmable logic. With gate densities approaching those of gate arrays and a reconfigurable architecture, the logic designer is freed from choosing between scores of generic programmable logic to perhaps find an acceptable match for his or her design needs. Adding to the list of benefits is the fact that the 5C121 is erasable. Now sections of the design can actually be programmed and tested in the device — without sacrificing a part to the circular file. In addition, there is no longer a need to generate test vectors to qualify the programming of the parts. EPLDs are erasable and therefore 100% testable at the factory.

OBJECTIVE

The purpose of this application note is to demonstrate the architectural options of the 5C121 by designing a digital crosspoint switch. Conceptually, a digital crosspoint switch switches data from any input to any output. Figure 1 shows a block diagram of a bytewide crosspoint switch.

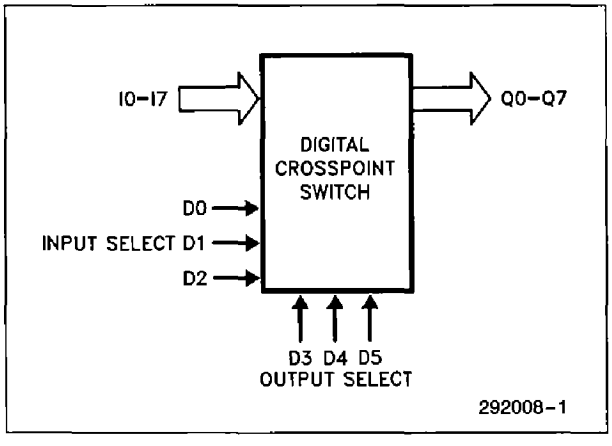

Figure 1. Functional Diagram of a Digital Crosspoint Switch

This design will employ features such as: registered output with registered feedback, combinational feedback, input latches, buried registers, and dual clock options. The digital crosspoint switch in this design can route data from one of eight inputs to one of eight outputs in a single clock cycle. Options for holding the deselected outputs at previous levels, latching inputs, and fitting considerations are explored.

THE BASIC ARCHITECTURE

The 5C121 contains 28 Macrocells, 12 dedicated inputs, 24 programmable I/O lines, and two clocks input pins. Inputs may be flow through, or latched on the rising or falling edge of either clock. Output options include registered or combinational output. In addition, each output may be fed back into the array in both the true and complement version. For a more complete description of the 5C121 architecture the reader is referred to the 5C121 data sheet.

COMBINATIONAL FEEDBACK

Feedback in logic designs is used for a variety of reasons. Combinational feedback in the 5C121 is often used to reduce the number of product terms feeding one Macrocell. Though the 5C121 has Macrocells that can accept up to 16 product terms, all Macrocells are not that wide.

Let's look at an example. Equation 1 represents one of the eight Boolean expressions necessary to implement a digital crosspoint switch. Logically, this expression selects one of eight input signals (I0–I7), and routes that signal to Q0. Data bits D0, D1, and D2 select one of the eight input lines. In this case, data bits !D3, !D4, and !D5 select output Q0. (The exclamation point is used to indicate a logical complement of the signal.) Equations for Q1 through Q7 are very similar and will be discussed later.

```
Q0 = ( I0 × !D2 × !D1 × !D0
     + I1 × !D2 × !D1 × D0
     + I2 × !D2 × D1 × !D0
     + I3 × !D2 × D1 × D0
     + I4 × D2 × !D1 × !D0
     + I5 × D2 × !D1 × D0
     + I6 × D2 × D1 × !D0
     + I7 × D2 × D1 × D0 ) × !D5 × !D4 × !D3;  (1)

SELECTEQ = I0 × !D2 × !D1 × !D0
         + I1 × !D2 × !D1 × D0
         + I2 × !D2 × D1 × !D0
         + I3 × !D2 × D1 × D0
         + I4 × D2 × !D1 × !D0
         + I5 × D2 × !D1 × D0
         + I6 × D2 × D1 × !D0
         + I7 × D2 × D1 × D0;                (2)
```

Equation 2 contains the terms that will be common to all eight output equations. Both equations in this case contain eight product terms. By treating equation 2 as one common signal and routing that signal through combinational feedback, we can reduce the number of product terms in equations Q0 thru Q7 to one p-term each. The advantage is that the outputs can now be placed in any of the 24 I/O Macrocells available in the 5C121. In addition, the 5C121 contains four buried registers. (Buried registers have no output and are used solely for feedback.) If a buried register is available, iPLDs (Intel's Programmable Logic Development System) will automatically assign the No Output — Combinational Feedback function to a buried register. This increases the flexibility for pin assignments and makes

COMBINATIONAL FEEDBACK
(Continued)

p-terms available in case a design change is needed. Equations 3 thru 10 reflect this improvement.

Q0 = SELECTEQ × !D5 × !D4 × !D3; (3)

Q1 = SELECTEQ × !D5 × !D4 × D3; (4)

Q2 = SELECTEQ × !D5 × D4 × !D3; (5)

Q3 = SELECTEQ × !D5 × D4 × D3; (6)

Q4 = SELECTEQ × D5 × !D4 × !D3; (7)

Q5 = SELECTEQ × D5 × !D4 × D3; (8)

Q6 = SELECTEQ × D5 × D4 × !D3; (9)

Q7 = SELECTEQ × D5 × D4 × D3; (10)

REGISTERED FEEDBACK

Registered feedback is also employed in a variety of applications such as counters and state machines. In this particular example, the registered feedback signal can be used to hold the deselected outputs of the switch at their previous level until that output is selected again. This is accomplished by simply "ANDing" the feedback signal with the inversion of the output select signal. The result is then "ORed" with the equation for the given output. Holding the previous output might be useful in control applications or when interfacing to slow peripherals. Equations 11 thru 18 are the result.

Q0 = SELECTEQ × !D5 × !D4 × !D3 + !(D5 × !D4 × !D3) × Q0—fdbk; (11)

Q1 = SELECTEQ × !D5 × !D4 × D3 + !(!D5 × !D4 × D3) × Q1—fdbk; (12)

Q2 = SELECTEQ × !D5 × D4 × !D3 + !(!D5 × D4 × !D3) × Q2—fdbk; (13)

Q3 = SELECTEQ × !D5 × D4 × D3 + !(!D5 × D4 × D3) × Q3—fdbk; (14)

Q4 = SELECTEQ × D5 × !D4 × D3 + !(D5 × !D4 × !D3) × Q4—fdbk; (15)

Q5 = SELECTEQ × D5 × !D4 × D3 + !(D5 × !D4 × D3) × Q5—fdbk; (16)

Q6 = SELECTEQ × D5 × D4 × !D3 + !(D5 × D4 × !D3) × Q6—fdbk; (17)

Q7 = SELECTEQ × D5 × DR × D3 + !(D5 × D4 × DE) × Q7—fdbk; (18)

Equations 11 thru 18 are all that are necessary to implement a digital crosspoint switch with the output hold feature. Each equation contains only four product terms when written in the expanded form and could therefore fit into any Macrocell in the 5C121. The appendix contains the report and ADF files generated by the iPLDs software.

TIMING ANALYSIS

Figure 2 shows the internal delay paths associated with this design in the 5C121. The frequency at which the 5C121 may be clocked can be determined by examining the internal delay elements of the 5C121. These include the input delay (Tin), two array delays (Tad), and the combinational feedback delay (Tcf). Table 1 gives the simulation data for each of these paths in a 5C121-50.

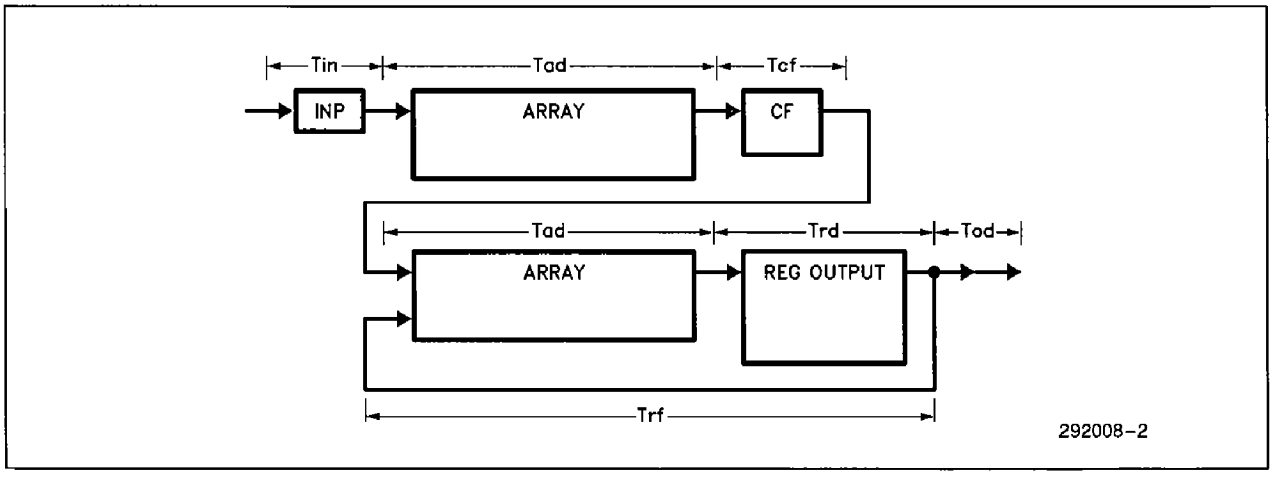

Figure 2. Crosspoint Delay Path

TIMING ANALYSIS (Continued)

Table 1. 5C121-50 Simulation Data

Model Parameter	Delay (ns)
Tad	38
Trd	7
Tod	8
Tin	10
Tic	8
Trf	5
Tcf	5

The sum of the delays before the register input equal the set-up time Tsu with reference to the internal clock. By substracting the input clock delay Tic we shift the reference to the external clock pin. The set-up time with reference to external signals is shown in equation 19. Inverting this signal yields the maximum clock frequency, fmax. The maximum clock frequency is shown in equation 20.

$$Tsu = Tin + 2Tad + Tcf - Tic; \quad (19)$$

$$fmax = 1\ Tsu \quad (20)$$

Therefore, this configuration of the 5C121-50 could be clocked at 10 MHz, allowing a data transfer rate of 10 Mbits/second. By paralleling six 5C121s together, eight bits could be switched per cycle. Figure 3 shows the timing diagram for this configuration of the 5C121 digital crosspoint switch. Included in the appendix is the Advanced Design File (ADF), Logic Equation File (LEF), and Utilization report generated by Intel's Programmable Logic Software (iPLS) for this design.

INPUT LATCHES

One point must be raised about Figure 3. Notice that the time allowed for external data set-up is only 17 ns. Therefore, 17 ns after the rising edge of the clock, data must be stable and remain stable at the input pins until the next clock pulse. In most systems this would be a very stringent requirement. Fortunately the 5C121 has the ability to latch the data at the input pins with 7475 type transparent latches. Employing this feature eases the data set-up requirement as shown in Figure 4.

SUMMARY

The flexible architecture of the 5C121 gives the designer a variety of options for input and output configurations. Inputs may be latched to ease system timing requirements. Outputs may be clocked for synchronous systems or fed directly out as asynchronous signals. Feedback can be used to reduce product term requirements, to save present state information for state machines and counters, or simply to hold deselected outputs as shown in this example. Imagine the possibilities.

J. R. Donnell
PLDO Applications

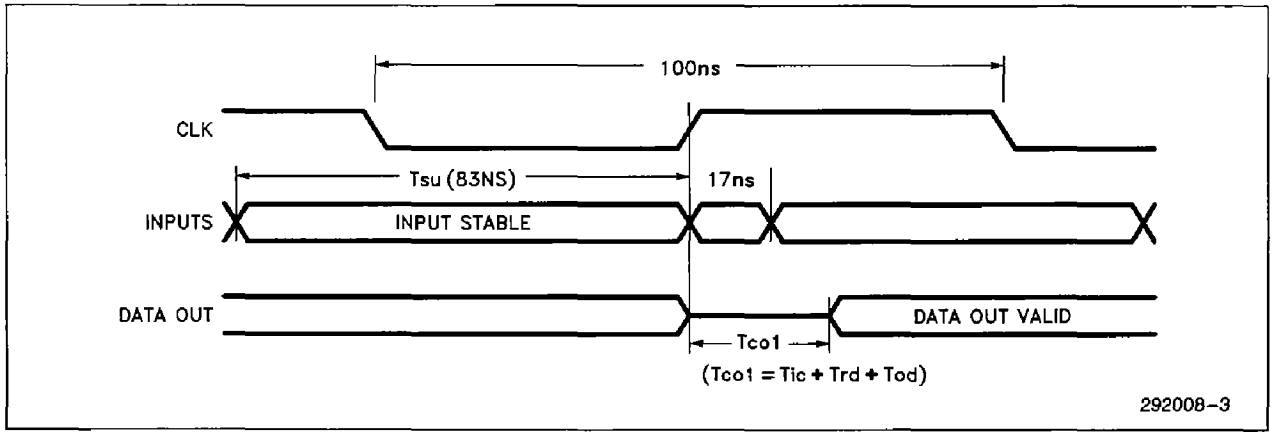

Figure 3. Crosspoint Timing Diagram

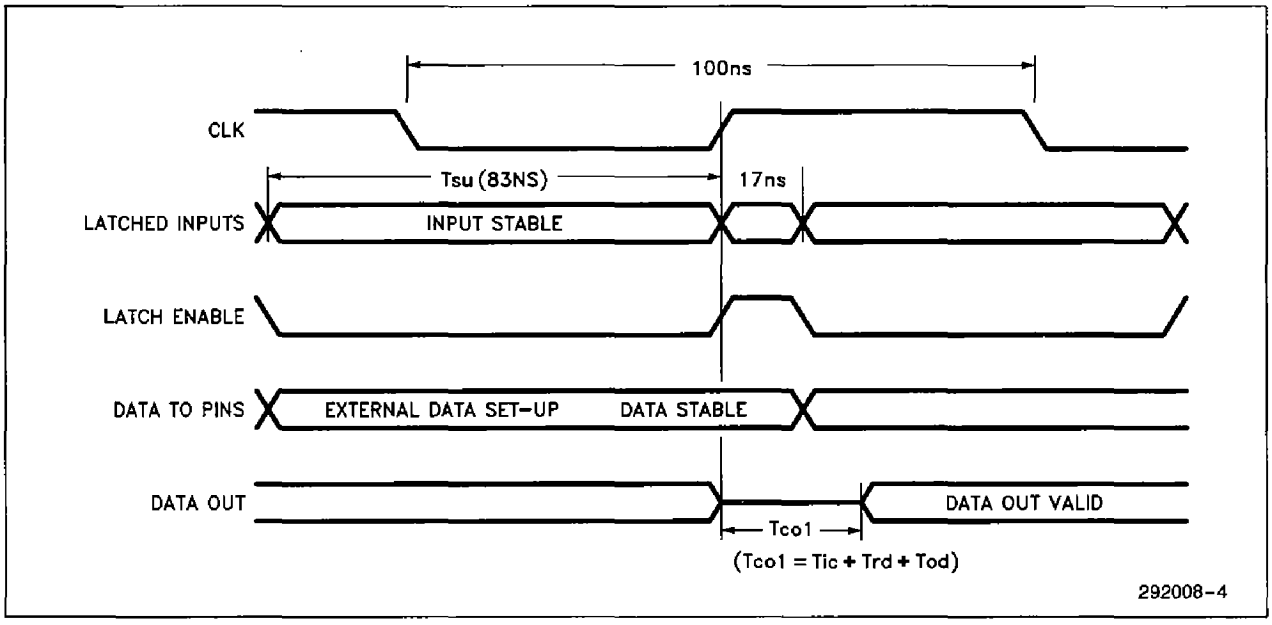

Figure 4. Crosspoint Timing Diagram with Input Latches

APPENDIX

ADF File

```
0
5C121
Digital Crosspoint Switch
LB Version 3.0, Baseline 17x, 9/26/85
PART: 5C121
INPUTS: I00@37,I01@36,I02@35,I03@34,I04@8,I05@9,I06@10,I07@11,I10@33,I11@32
       ,I12@31,I13@30,I14@29,I15@28,I16@27,I17@26,CLK@38,D0@2,D1@3,D2@4,D3@5
       ,D4@6,D5@7,ILE@1
OUTPUTS: Q00@12,Q01@13,Q02@14,Q03@15,Q04@16,Q05@17,Q06@18,Q07@19,Q10@24,Q11@23
        ,Q12@22,Q13@21
NETWORK:
Q00,Q00FBK = RORF (Q00D,CLK,GND,GND,VCC) % BIT 0 OUTPUTS %
Q01,Q01FBK = RORF (Q01D,CLK,GND,GND,VCC)
Q02,Q02FBK = RORF (Q02D,CLK,GND,GND,VCC)
Q03,Q03FBK = RORF (Q03D,CLK,GND,GND,VCC)
Q04,Q04FBK = RORF (Q04D,CLK,GND,GND,VCC)
Q05,Q05FBK = RORF (Q05D,CLK,GND,GND,VCC)
Q06,Q06FBK = RORF (Q06D,CLK,GND,GND,VCC)
Q07,Q07FBK = RORF (Q07D,CLK,GND,GND,VCC)
Q10,Q10FBK = RORF (Q10D,CLK,GND,GND,VCC) % 4 OF THE 8, BIT 0 OUTPUTS%
Q11,Q11FBK = RORF (Q11D,CLK,GND,GND,VCC)
Q12,Q12FBK = RORF (Q12D,CLK,GND,GND,VCC)
Q13,Q13FBK = RORF (Q13D,CLK,GND,GND,VCC)
CLK = INP (CLK)
D5 = LINP (D5,ILE) % OUTPUT SELECT CONTROL BITS %
ILE = INP (ILE)
D4 = LINP (D4,ILE)
D3 = LINP (D3,ILE)
D2 = LINP (D2,ILE) % INPUT SELECT CONTROL BITS %
D1 = LINP (D1,ILE)
D0 = LINP (D0,ILE)
I00 = LINP (I00,ILE)
I01 = LINP (I01,ILE)
I02 = LINP (I02,ILE)
I03 = LINP (I03,ILE)
I04 = LINP (I04,ILE)
I05 = LINP (I05,ILE)
I06 = LINP (I06,ILE)
I07 = LINP (I07,ILE)
I10 = LINP (I10,ILE) % INPUTS FOR BIT 1 SWITCH %
I11 = LINP (I11,ILE)
I12 = LINP (I12,ILE)
I13 = LINP (I13,ILE)
I14 = LINP (I14,ILE)
I15 = LINP (I15,ILE)
I16 = LINP (I16,ILE)
I17 = LINP (I17,ILE)
SELECTEQ0F = NOCF (SELECTEQ0)
SELECTEQ1F = NOCF (SELECTEQ1)
EQUATIONS:
Q00D = SELECTEQ0F*!D5*!D4*!D3
        + !(!D5*!D4*!D3)*Q00FBK;
Q01D = SELECTEQ0F*!D5*!D4* D3
        + !(!D5*!D4* D3)*Q01FBK;
Q02D = SELECTEQ0F*!D5* D4*!D3
        + !(!D5* D4*!D3)*Q02FBK;
Q03D = SELECTEQ0F*!D5* D4* D3
        + !(!D5* D4* D3)*Q03FBK;
Q04D = SELECTEQ0F* D5*!D4*!D3
        + !( D5*!D4*!D3)*Q04FBK;
Q05D = SELECTEQ0F* D5*!D4* D3
```

292008-5

ADF File (Continued)

```
        + !( D5*!D4* D3)*Q05FBK;
Q06D = SELECTEQ0F* D5* D4*!D3
        + !( D5* D4*!D3)*Q06FBK;
Q07D = SELECTEQ0F* D5* D4* D3
        + !( D5* D4* D3)*Q07FBK;
Q10D = SELECTEQ1F*!D5*!D4*!D3
        + !(!D5*!D4*!D3)*Q10FBK;
Q11D = SELECTEQ1F*!D5*!D4* D3
        + !(!D5*!D4* D3)*Q11FBK;
Q12D = SELECTEQ1F*!D5* D4*!D3
        + !(!D5* D4*!D3)*Q12FBK;
Q13D = SELECTEQ1F*!D5* D4* D3
        + !(!D5* D4* D3)*Q13FBK;
SELECTEQ0 = I00*!D2*!D1*!D0      % COMMON EQUATION FOR BIT 0 %
        + I01*!D2*!D1*D0
        + I02*!D2*D1*!D0
        + I03*!D2*D1*D0
        + I04*D2*!D1*!D0
        + I05*D2*!D1*D0
        + I06*D2*D1*!D0
        + I07*D2*D1*D0;
SELECTEQ1 = I10*!D2*!D1*!D0      % COMMON EQUATION FOR BIT 1 %
        + I11*!D2*!D1*D0
        + I12*!D2*D1*!D0
        + I13*!D2*D1*D0
        + I14*D2*!D1*!D0
        + I15*D2*!D1*D0
        + I16*D2*D1*!D0
        + I17*D2*D1*D0;
END$
```

292008-6

LEF File

```
JR Donnell
Intel
January 24, 1986

0
5C121
Digital Crosspoint Switch
LB Version 3.0, Baseline 17x, 9/26/85
PART:
        5C121
INPUTS:
        I00@37, I01@36, I02@35, I03@34, I04@8, I05@9, I06@10, I07@11, I10@33,
        I11@32, I12@31, I13@30, I14@29, I15@28, I16@27, I17@26, CLK@38, D0@2,
        D1@3, D2@4, D3@5, D4@6, D5@7, ILE@1
OUTPUTS:
        Q00@12, Q01@13, Q02@14, Q03@15, Q04@16, Q05@17, Q06@18, Q07@19, Q10@24,
        Q11@23, Q12@22, Q13@21
NETWORK:
        CLK = INP(CLK)
        ILE = INP(ILE)
        I00 = LINP(I00, ILE)
        I01 = LINP(I01, ILE)
        I02 = LINP(I02, ILE)
        I03 = LINP(I03, ILE)
        I04 = LINP(I04, ILE)
        I05 = LINP(I05, ILE)
        I06 = LINP(I06, ILE)
        I07 = LINP(I07, ILE)
        I10 = LINP(I10, ILE)
        I11 = LINP(I11, ILE)
        I12 = LINP(I12, ILE)
        I13 = LINP(I13, ILE)
        I14 = LINP(I14, ILE)
        I15 = LINP(I15, ILE)
        I16 = LINP(I16, ILE)
        I17 = LINP(I17, ILE)
        D0 = LINP(D0, ILE)
        D1 = LINP(D1, ILE)
        D2 = LINP(D2, ILE)
        D3 = LINP(D3, ILE)
        D4 = LINP(D4, ILE)
        D5 = LINP(D5, ILE)
        Q00, Q00FBK = RORF(Q00D, CLK, GND, GND, VCC)
        Q01, Q01FBK = RORF(Q01D, CLK, GND, GND, VCC)
        Q02, Q02FBK = RORF(Q02D, CLK, GND, GND, VCC)
        Q03, Q03FBK = RORF(Q03D, CLK, GND, GND, VCC)
        Q04, Q04FBK = RORF(Q04D, CLK, GND, GND, VCC)
        Q05, Q05FBK = RORF(Q05D, CLK, GND, GND, VCC)
        Q06, Q06FBK = RORF(Q06D, CLK, GND, GND, VCC)
        Q07, Q07FBK = RORF(Q07D, CLK, GND, GND, VCC)
        Q10, Q10FBK = RORF(Q10D, CLK, GND, GND, VCC)
        Q11, Q11FBK = RORF(Q11D, CLK, GND, GND, VCC)
        Q12, Q12FBK = RORF(Q12D, CLK, GND, GND, VCC)
        Q13, Q13FBK = RORF(Q13D, CLK, GND, GND, VCC)
        SELECTEQ0F = NOCF(SELECTEQ0)
        SELECTEQ1F = NOCF(SELECTEQ1)
EQUATIONS:
        SELECTEQ1 = I10 * D2' * D1' * D0'
                  + D2 * D1' * D0' * I14
                  + D2' * D1 * D0' * I12
                  + D2' * D1' * D0 * I11
                  + D2 * D1 * D0' * I16
                  + D2 * D1' * D0 * I15
                  + D2' * D1 * D0 * I13
```

292008-12

LEF File (Continued)

```
                + D2 * D1 * D0 * I17;

        SELECTEQ0 = I00 * D2' * D1' * D0'
                + D2 * D1' * D0' * I04
                + D2' * D1 * D0' * I02
                + D2' * D1' * D0 * I01
                + D2 * D1 * D0' * I06
                + D2 * D1' * D0 * I05
                + D2' * D1 * D0 * I03
                + D2 * D1 * D0 * I07;

        Q13D = D3' * Q13FBK
             + D4' * Q13FBK
             + D5 * Q13FBK
             + SELECTEQ1F * D5' * D4 * D3;

        Q12D = D4' * Q12FBK
             + D3 * Q12FBK
             + D5 * Q12FBK
             + SELECTEQ1F * D5' * D4 * D3';

        Q11D = D3' * Q11FBK
             + D4 * Q11FBK
             + D5 * Q11FBK
             + SELECTEQ1F * D5' * D4' * D3;

        Q10D = D3 * Q10FBK
             + D4 * Q10FBK
             + D5 * Q10FBK
             + SELECTEQ1F * D5' * D4' * D3';

        Q07D = D3' * Q07FBK
             + D4' * Q07FBK
             + D5' * Q07FBK
             + SELECTEQ0F * D5 * D4 * D3;

        Q06D = D4' * Q06FBK
             + D5' * Q06FBK
             + D3 * Q06FBK
             + SELECTEQ0F * D5 * D4 * D3';

        Q05D = D3' * Q05FBK
             + D5' * Q05FBK
             + D4 * Q05FBK
             + SELECTEQ0F * D5 * D4' * D3;

        Q04D = D5' * Q04FBK
             + D3 * Q04FBK
             + D4 * Q04FBK
             + SELECTEQ0F * D5 * D4' * D3';

        Q03D = D3' * Q03FBK
             + D4' * Q03FBK
             + D5 * Q03FBK
             + SELECTEQ0F * D5' * D4 * D3;

        Q02D = D4' * Q02FBK
             + D3 * Q02FBK
             + D5 * Q02FBK
             + SELECTEQ0F * D5' * D4 * D3';

        Q01D = D3' * Q01FBK
             + D4 * Q01FBK
             + D5 * Q01FBK
             + SELECTEQ0F * D5' * D4' * D3;

        Q00D = D3 * Q00FBK
             + D4 * Q00FBK
             + D5 * Q00FBK
             + SELECTEQ0F * D5' * D4' * D3';

END$
```

292008-13

292008-14

RPT File

```
Logic Optimizing Compiler Utilization Report

 ***** Design implemented successfully

JR Donnell
Intel
January 24, 1986

0
5C121
Digital Crosspoint Switch
LB Version 3.0, Baseline 17x, 9/26/85

                5C121
              - - - - -
          ILE -| 1   40|- Vcc
           D0 -| 2   39|- Vcc
           D1 -| 3   38|- CLK
           D2 -| 4   37|- I00
           D3 -| 5   36|- I01
           D4 -| 6   35|- I02
           D5 -| 7   34|- I03
          I04 -| 8   33|- I10
          I05 -| 9   32|- I11
          I06 -|10   31|- I12
          I07 -|11   30|- I13
          Q00 -|12   29|- I14
          Q01 -|13   28|- I15
          Q02 -|14   27|- I16
          Q03 -|15   26|- I17
          Q04 -|16   25|- GND
          Q05 -|17   24|- Q10
          Q06 -|18   23|- Q11
          Q07 -|19   22|- Q12
          GND -|20   21|- Q13
               - - - - -


**INPUTS**

                                                    |          Feeds:
      Name  Pin  Resource   MCell #    PTerms       |  MCells     OE  Clear  Clock

       ILE    1       INP         -         -       |        -      -      -  Latch

        D0    2      LINP         -         -       |       13      -      -      -
                                                    |       15

        D1    3      LINP         -         -       |       13      -      -      -
                                                    |       15

        D2    4      LINP         -         -       |       13      -      -      -
                                                    |       15

        D3    5      LINP         -         -       |        9      -      -      -
                                                    |       10
                                                    |       11
                                                    |       12
                                                    |       17
                                                    |       18
                                                    |       19
                                                    |       20
                                                    |       21
```

292008-9

RPT File (Continued)

					22 23 24			
D4	6	LINP	-	-	9 10 11 12 17 18 19 20 21 22 23 24	-	-	-
D5	7	LINP	-	-	9 10 11 12 17 18 19 20 21 22 23 24	-	-	-
I04	8	LINP	28	0/ 4	15	-	-	-
I05	9	LINP	27	0/10	15	-	-	-
I06	10	LINP	26	0/ 8	15	-	-	-
I07	11	LINP	25	0/ 6	15	-	-	-
I17	26	LINP	7	0/10	13	-	-	-
I16	27	LINP	6	0/ 8	13	-	-	-
I15	28	LINP	5	0/ 6	13	-	-	-
I14	29	LINP	4	0/ 6	13	-	-	-
I13	30	LINP	3	0/ 8	13	-	-	-
I12	31	LINP	2	0/10	13	-	-	-
I11	32	LINP	1	0/ 4	13	-	-	-
I10	33	LINP	-	-	13	-	-	-
I03	34	LINP	-	-	15	-	-	-
I02	35	LINP	-	-	15	-	-	-
I01	36	LINP	-	-	15	-	-	-
I00	37	LINP	-	-	15	-	-	-
CLK	38	INP	-	-	-	-	-	Reg

292008-10

RPT File (Continued)

```
**OUTPUTS**

                                                          Feeds:
      Name   Pin   Resource   MCell #     PTerms    MCells     OE   Clear

       Q00    12       RORF        24       4/ 6        24      -       -
       Q01    13       RORF        23       4/ 8        23      -       -
       Q02    14       RORF        22       4/10        22      -       -
       Q03    15       RORF        21       4/ 4        21      -       -
       Q04    16       RORF        20       4/12        20      -       -
       Q05    17       RORF        19       4/ 4        19      -       -
       Q06    18       RORF        18       4/ 8        18      -       -
       Q07    19       RORF        17       4/ 8        17      -       -
       Q13    21       RORF        12       4/ 8        12      -       -
       Q12    22       RORF        11       4/ 8        11      -       -
       Q11    23       RORF        10       4/ 4        10      -       -
       Q10    24       RORF         9       4/12         9      -       -


**BURIED REGISTERS**

                                                          Feeds:
      Name   Pin   Resource   MCell #     PTerms    MCells     OE   Clear

               -       NOCF        13       8/ 8         9      -       -
                                                        10
                                                        11
                                                        12

               -       NOCF        15       8/ 8        17      -       -
                                                        18
                                                        19
                                                        20
                                                        21
                                                        22
                                                        23
                                                        24


**UNUSED RESOURCES**

      Name   Pin   Resource      MCell     PTerms

         -    25          -          8          4
         -    NA          -         14          8
         -    NA          -         16          8

**PART UTILIZATION**

97%      Pins
89%      MacroCells
30%      Pterms
```

292008-11

INTEL CORPORATION, 3065 Bowers Ave., Santa Clara, CA 95051; Tel. (408) 987-8080

INTEL CORPORATION (U.K.) Ltd., Swindon, United Kingdom; Tel. (793) 696 000

INTEL JAPAN k.k., Ibaraki-ken; Tel. 029747-8511

Printed in U.S.A./P-125/0486/7K/BL/LK
Memory Components